

Doc. no: N0868 / 96-0050  
Date: 25 January, 1996  
Project: Programming Language C++  
Reply to: T. Kamimura  
E-mail: kamimura@trl.ibm.co.jp

## Japanese Comments on "Extended Characters in C++ Programs"

Tom Plum and Dag Bruck have proposed to support extended characters in Document X3J16/95-0220(WG21/N0820). This memo is to express a view shared by Japanese members of WG21 on the proposal.

The proposal calls for the support of ISO Latin-1 encoding for source files, and for the support of all characters representable in ISO 10646. In addition, it proposes the introduction of trigraphs for characters in ISO 10646 that are not representable in the source character set. It also supports the idea of extended identifier characters as being proposed as "International identifier characters" in ISO/IEC PDTR 10176. Also, it requires to define for each character in a literal or a string the implementation-defined character in the target character set.

These proposed items have significant implications on the current situation of character set and encoding environment. Some will be very difficult to support, and others will require detailed study for recommendations for practical migration. If the proposal is adopted without such careful analysis, the standard will not be implementable at least in Japan and will therefore become useless.

We believe that the character set and encoding are very delicate and complex issues. The current situation is realized as a balance of practical diversity, political climate among various competitors and international competition and collaboration, and basic ideas on which standards can practically be grounded.

The notions of two character sets, one using ISO 646 invariant set as a basic character set to support maximum portability, and an extended character set to support portability for regional/local/cultural needs created reasonable and practical environment. If a standard enforces the support of ISO 10646 for identifier characters, then this will likely change. We are aware that PDTR 10176 has a list of "international identifier characters" in its annex, but this is a controversial subject, and at least Japanese representatives of SC22/WG20 are not for full support at this stage.

Also, our understanding is that no programming language standard enforces one particular encoding scheme. This is important and practical as there are three major encoding schemes used currently in Japan: JIS (Japanese Industrial Standard), Shift JIS, and EUC encoding. Latin-1 is not compatible with any of these major encoding. For example, a single byte is used for Roman character and half width Katakana in Shift JIS has a direct conflict against Latin 1 encoding. If the standard enforces Latin 1 encoding, it will require major change on the current environment which is very unlikely to be accepted by our community.

To support a standard which will require changes of current environment, we need to formulate possible and practical approach for migration. To analyze possible approach to support ISO 10646 character set, we need to understand details of what is intended by "accepting source files using all characters representable in ISO 10646". What will be recommended action if a character is not uniquely representable in the target character set? If we need to preserve uniqueness, it may require to define some run-time representation with escape characters.

Even though the proposal indicates that it does not specify encoding for them, defining such run-time implementation in many systems without major change to existing systems will be difficult since they are based on existing encoding schemes mentioned above. If it requires the ability to assign literals to array of char/wchar, the difficulty will increase. Unless we specify where and how the use of ISO 10646 character set is permitted, we cannot proceed in forming possible approach to support the standard.

The proposed issues are all important ones, and they will require careful analysis in each national environment. They are definitely not the issues we can decide in a single meeting or two. Considering the current schedule of our standardization process, we are wondering if we should investigate these issues further at this stage.